BRAIN TUMOR SEGMENTATION

Farida Ahmed Madkour
faridamadkour@aucegypt.edu
SID: 900211360

Mona Mahmoud Ibrahim
monamahmoud@aucegypt.edu
SID: 900212749

Sama Amr Gouda
samaamr@aucegypt.edu
SID: 900211296

TABLE OF CONTENTS

01

PROPOSAL SUMMARY & PROBLEM STATEMENT

PROBLEM STATEMENT

Brain tumor segmentation using MRI scans and detecting whether there is a tumor

CHOSEN DATASET

Brain Tumor Segmentation (BraTS) 2020

369 Brain MRI Images → 155 Image Slices per Patient

57195 Images and 4 Primary Files

BraTS20 Training Metadata

Slice Path, Target, Volume, Slice, and Channels

Survival Info

Age, Survival Days, and Extent of Resection

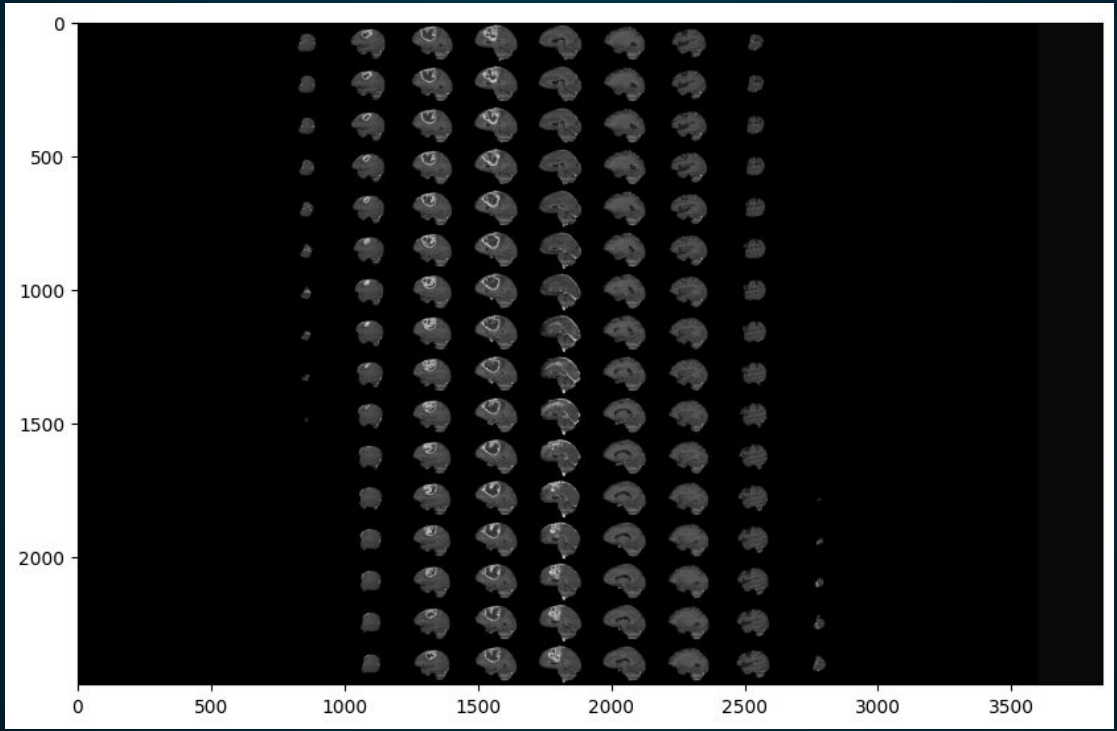
Name Mapping

77 Low-Grade Gliomas (LGG) Patients and 283 High-Grade Gliomas (HGG) Patients

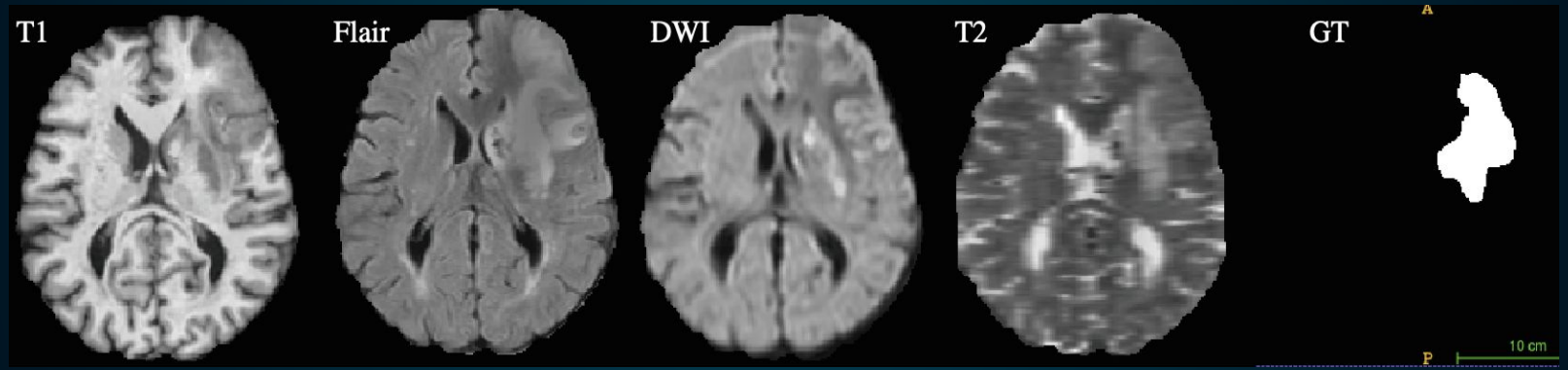
Meta Data

Slice Path, Target, Volume and Slice

CHOSEN DATASET CNTD.



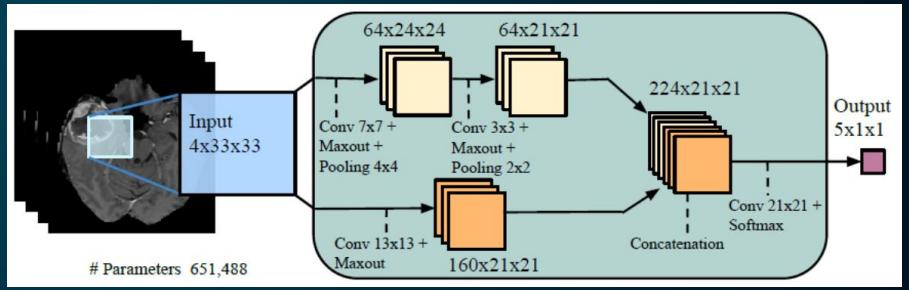
CHOSEN DATASET CNTD.



CHOSEN MODEL

TWO PATHS CNN → CASCADING ARCHITECTURE

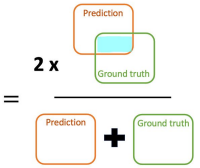
1. Two inputs are taken from the image
2. Inputs assembled after several activation functions
3. Output of one model be the input of the next one



EVALUATION METRICS

1. Dice Score → Dice Similarity Coefficient (DSC)
   a. Ranges from 0 to 1
   b. 0 = No Overlap
   c. 1 = Perfect Overlap

$$\text{Dice} = \frac{2\,X\,Area\,of\,overlap}{Total\,area} = \frac{2 \times}{\text{Prediction} + \text{Ground truth}}$$



2. Sensitivity (True Positive Rate or Recall)
   a. How well the model identifies all tumor voxels (true positives)
3. Accuracy
4. F1-Score
5. Loss
6. Precision

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

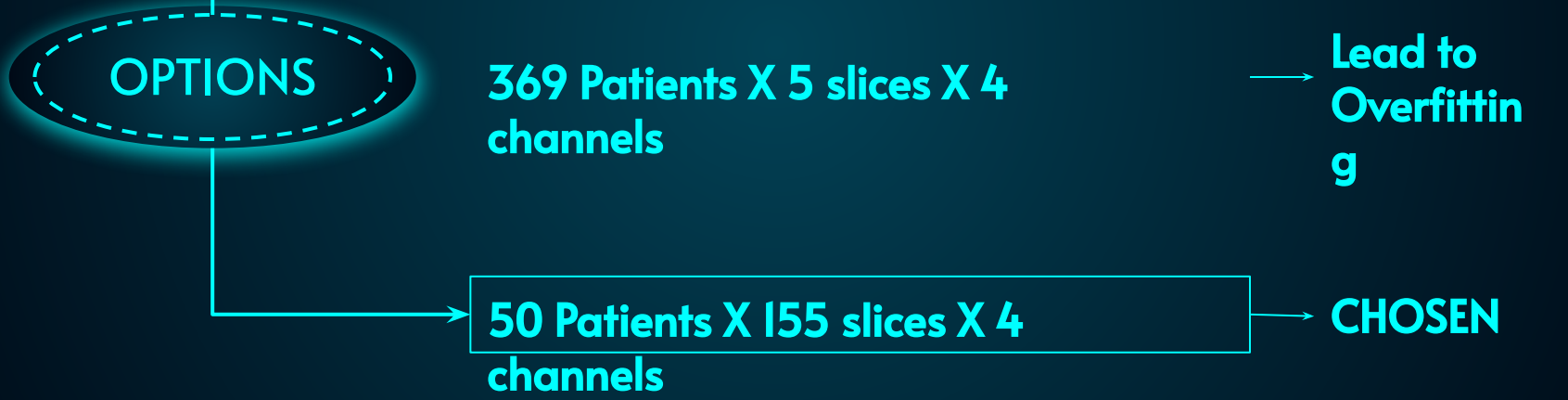
$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

02

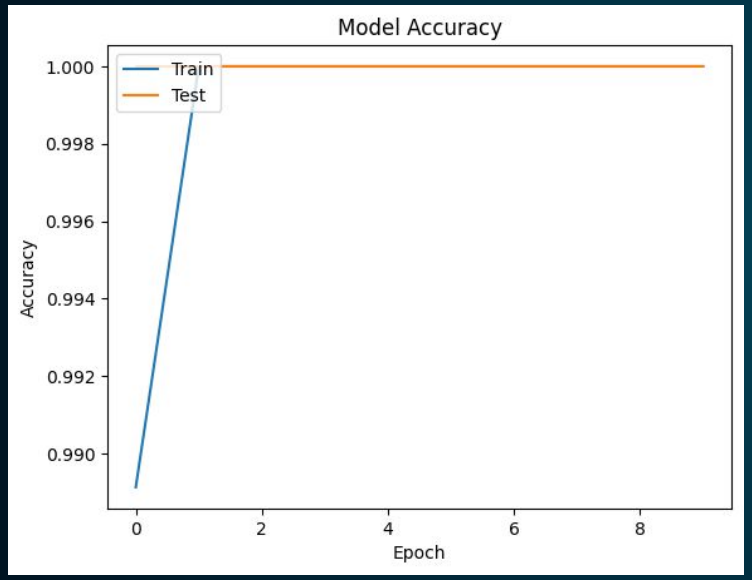
IMPLEMENTATION

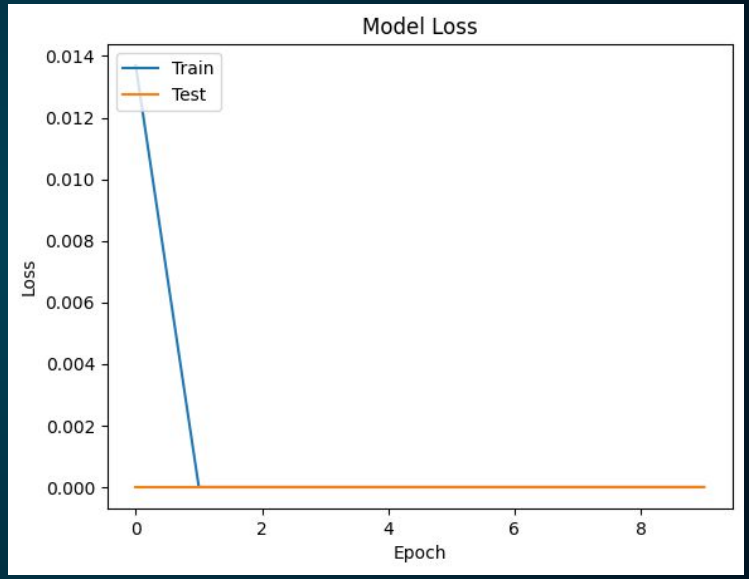
1. Skull-Stripping and Preprocessing
2. TwoPaths CNN with Attention Gates
3. U-Net
4. Ensemble of TwoPaths CNN and U-Net

PREPROCESSING

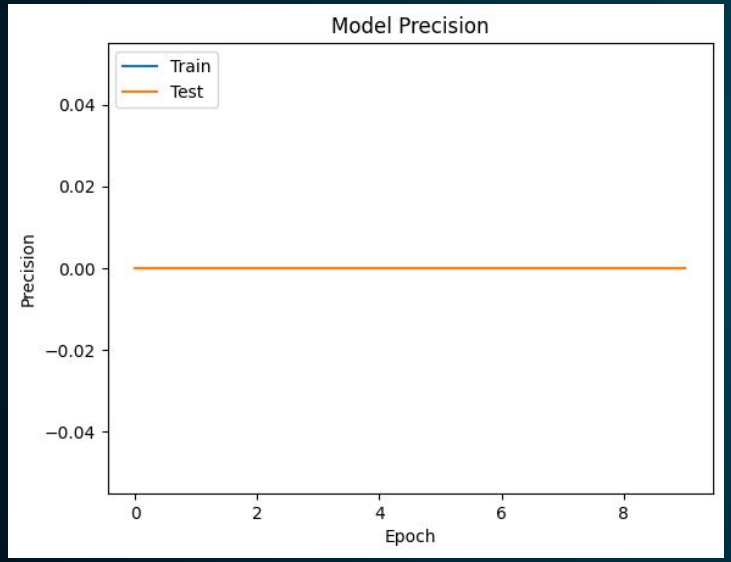


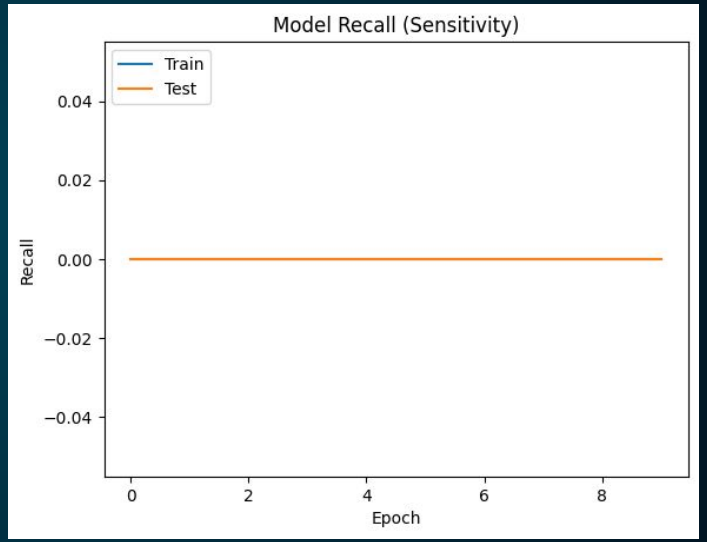
ATTEMPTS THAT LEAD TO OVERFITTING





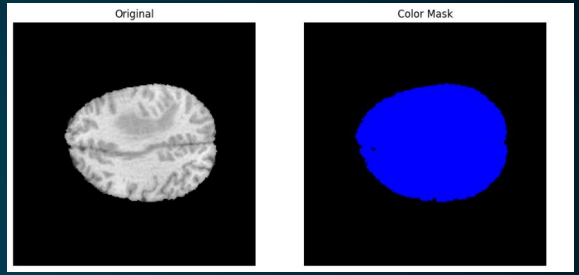
ATTEMPTS THAT LEAD TO OVERFITTING CNTD.

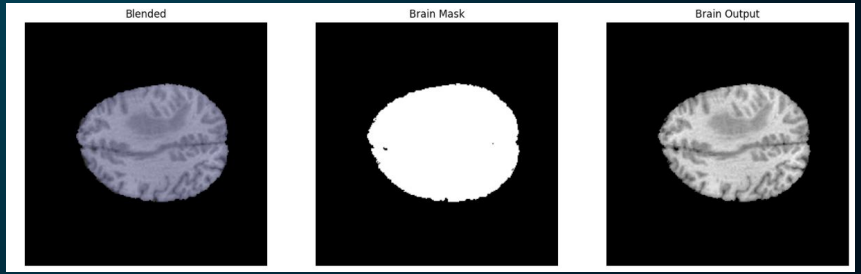




WHAT'S SKULL STRIPPING?

- Also known as brain extraction
- Removes non-brain areas before normalizing brains
- Useful for anonymizing brain scans
- Pre-processing step
- Improve model's robustness

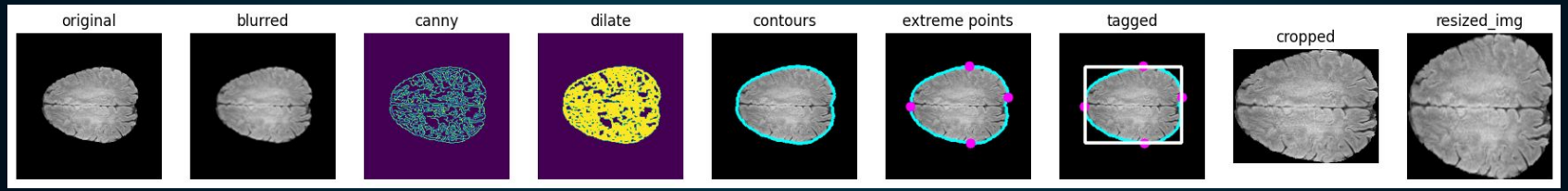




CANNY & CONTOUR EDGE DETECTION

- Detects abrupt local changes in the intensity of image
- Smooths the image to eliminate noise, and then it finds the image gradient

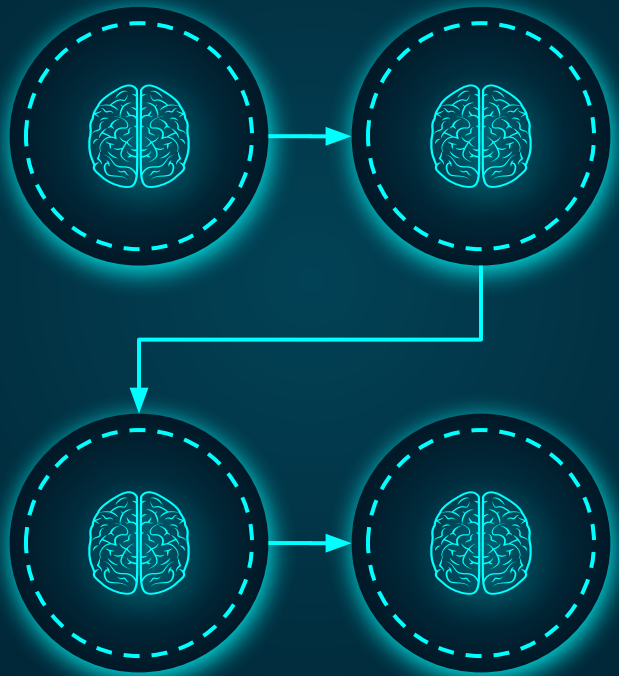
Strip The MRI → Blur → Canny Edge Detection → Identify Contours & Extreme Points → Crop Image → Resize (If necessary)



TWO PATHS CNN

TRAINING AND TESTING

80/20 SPLIT
Limitation due to size of data



RELU AND SIGMOID ACTIVATION FUNCTIONS IN DENSE LAYERS

SAME PADDING AND BATCH NORMALIZATION FOR MORE EFFECTIVE ANALYSIS

OPTIMIZING USING ADAM

TWO PATHS CNN CNTD.

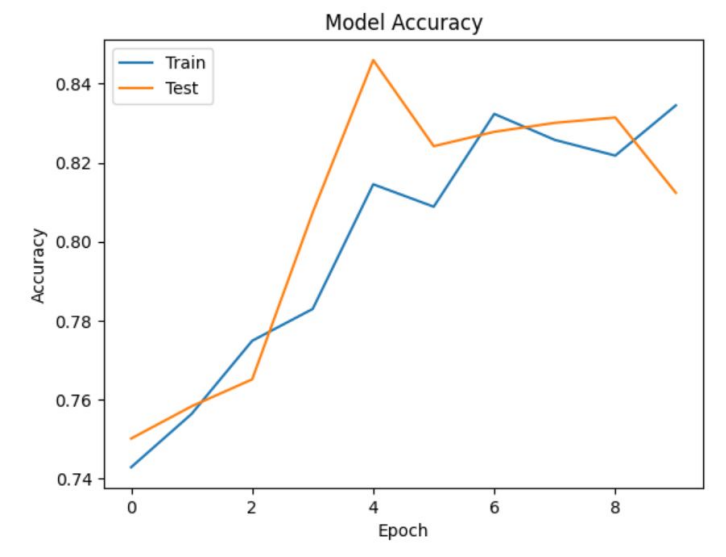
Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 240, 240, 4)	0	-
conv2d (Conv2D)	(None, 240, 240, 64)	12,608	input_layer[0][0]
conv2d_1 (Conv2D)	(None, 240, 240, 64)	12,608	input_layer[0][0]
batch_normalization (BatchNormalizatio…	(None, 240, 240, 64)	256	conv2d[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 64)	256	conv2d_1[0][0]
maximum (Maximum)	(None, 240, 240, 64)	0	batch_normalizat… batch_normalizat…
conv2d_2 (Conv2D)	(None, 240, 240, 64)	65,600	maximum[0][0]
conv2d_5 (Conv2D)	(None, 240, 240, 64)	36,928	conv2d_2[0][0]
conv2d_6 (Conv2D)	(None, 240, 240, 64)	36,928	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 240, 240, 160)	108,320	input_layer[0][0]
conv2d_4 (Conv2D)	(None, 240, 240, 160)	108,320	input_layer[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 64)	256	conv2d_5[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 64)	256	conv2d_6[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 160)	640	conv2d_3[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 160)	640	conv2d_4[0][0]
maximum_2 (Maximum)	(None, 240, 240, 64)	0	batch_normalizat… batch_normalizat…
maximum_1 (Maximum)	(None, 240, 240, 160)	0	batch_normalizat… batch_normalizat…
conv2d_7 (Conv2D)	(None, 240, 240, 64)	16,448	maximum_2[0][0]
concatenate (Concatenate)	(None, 240, 240, 224)	0	maximum_1[0][0], conv2d_7[0][0]
conv2d_8 (Conv2D)	(None, 240, 240, 2)	450	concatenate[0][0]
activation (Activation)	(None, 240, 240, 2)	0	conv2d_8[0][0]

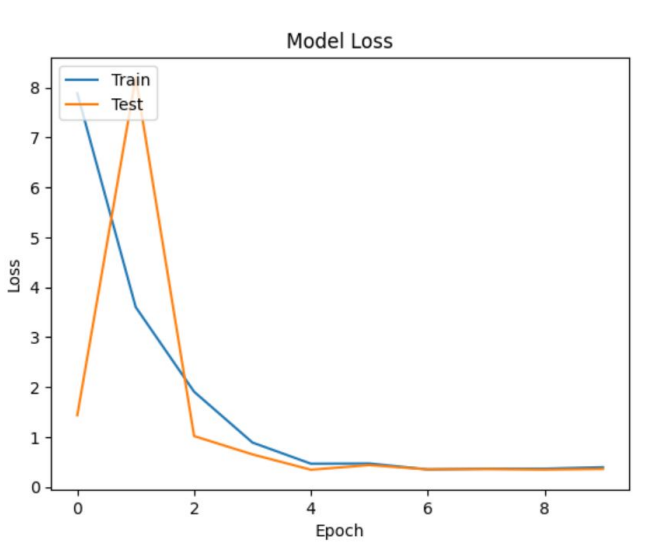
Total params: 400,514 (1.53 MB)

Trainable params: 399,362 (1.52 MB)

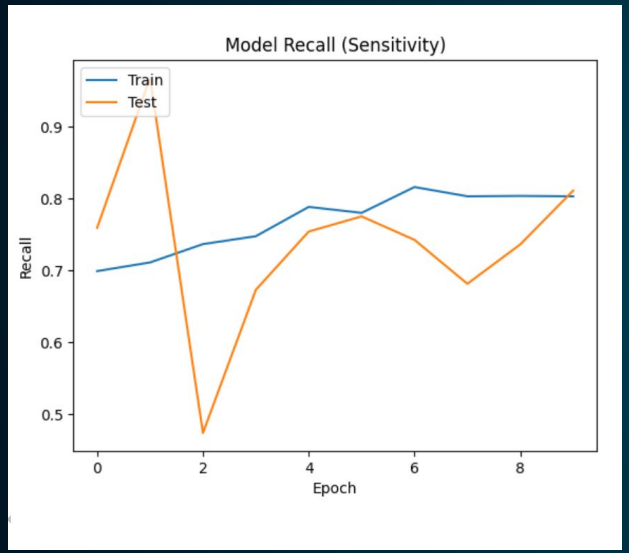
Non-trainable params: 1,152 (4.50 KB)

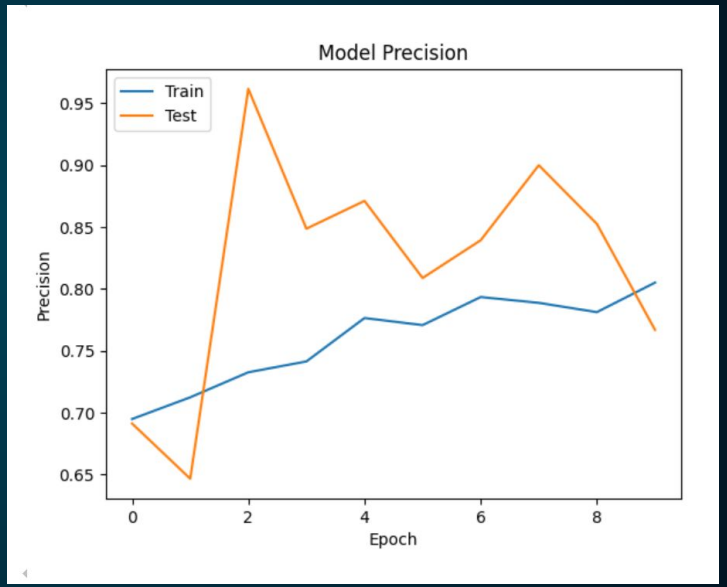
TWO PATHS CNN CNTD.





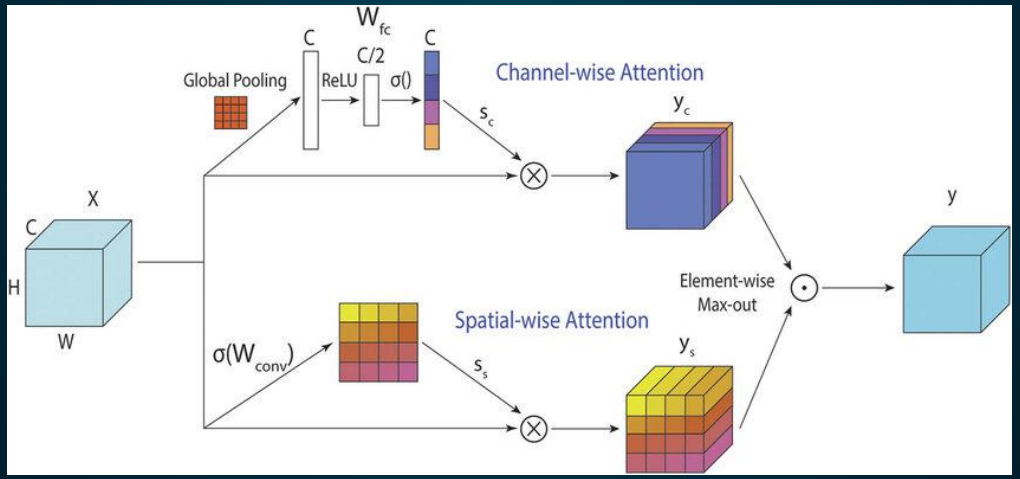
TWO PATHS CNN CNTD.





TWO PATHS CNN WITH ATTENTION GATES

1. For improved model performance
2. Focus on different regions of the image



TWO PATHS CNN WITH ATTENTION GATES CNTD.

Model: "functional_13"

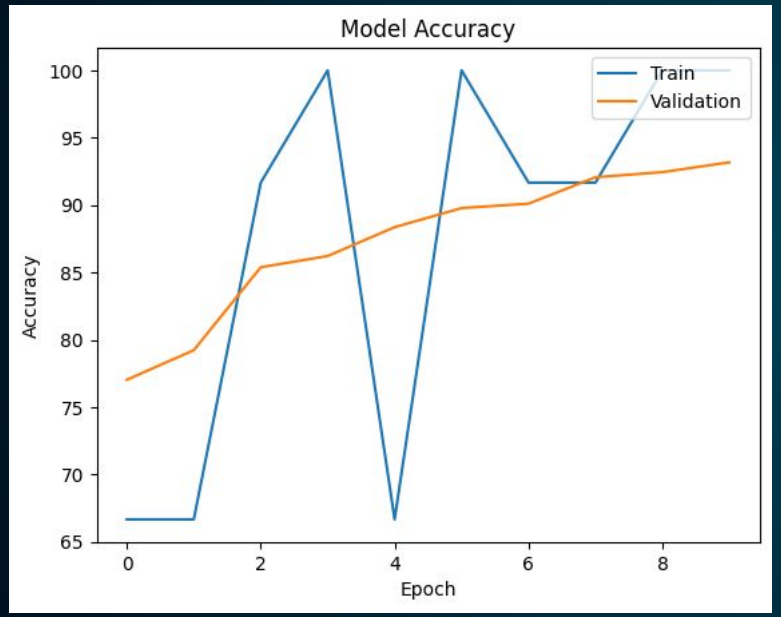
Layer (type)	Output Shape	Param #	Connected to
input_layer_12 (InputLayer)	(None, 240, 240, 4)	0	-
conv2d_110 (Conv2D)	(None, 240, 240, 64)	12,608	input_layer_12[0…
conv2d_111 (Conv2D)	(None, 240, 240, 64)	12,608	input_layer_12[0…
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 64)	256	conv2d_110[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 64)	256	conv2d_111[0][0]
conv2d_113 (Conv2D)	(None, 240, 240, 160)	108,320	input_layer_12[0…
conv2d_114 (Conv2D)	(None, 240, 240, 160)	108,320	input_layer_12[0…
maximum_36 (Maximum)	(None, 240, 240, 64)	0	batch_normalizat… batch_normalizat…
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 160)	640	conv2d_113[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 240, 240, 160)	640	conv2d_114[0][0]
conv2d_112 (Conv2D)	(None, 240, 240, 64)	65,600	maximum_36[0][0]
maximum_37 (Maximum)	(None, 240, 240, 160)	0	batch_normalizat… batch_normalizat…
bahdanau_attention… (BahdanauAttention)	[(None, 240, 240, 64), (None, None, 240, 240, 1)]	4,193	conv2d_112[0][0], conv2d_112[0][0]
bahdanau_attention… (BahdanauAttention)	[(None, 240, 240, 160), (None, None, 240, 240, 1)]	10,337	maximum_37[0][0], maximum_37[0][0]
concatenate_12 (Concatenate)	(None, 240, 240, 224)	0	bahdanau_attenti… bahdanau_attenti…
conv2d_118 (Conv2D)	(None, 240, 240, 2)	450	concatenate_12[0…
activation_12 (Activation)	(None, 240, 240, 2)	0	conv2d_118[0][0]
conv2d_119 (Conv2D)	(None, 240, 240, 1)	3	activation_12[0]…

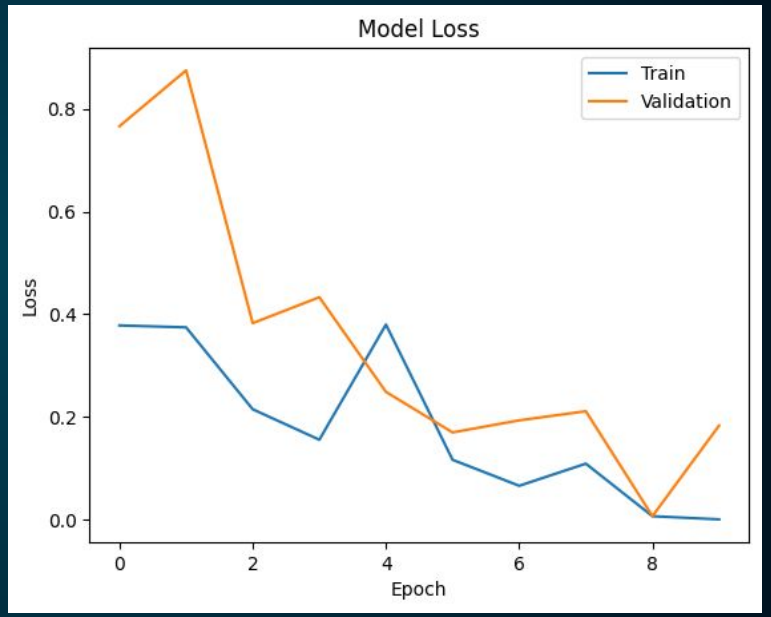
Total params: 324,231 (1.24 MB)

Trainable params: 323,335 (1.23 MB)

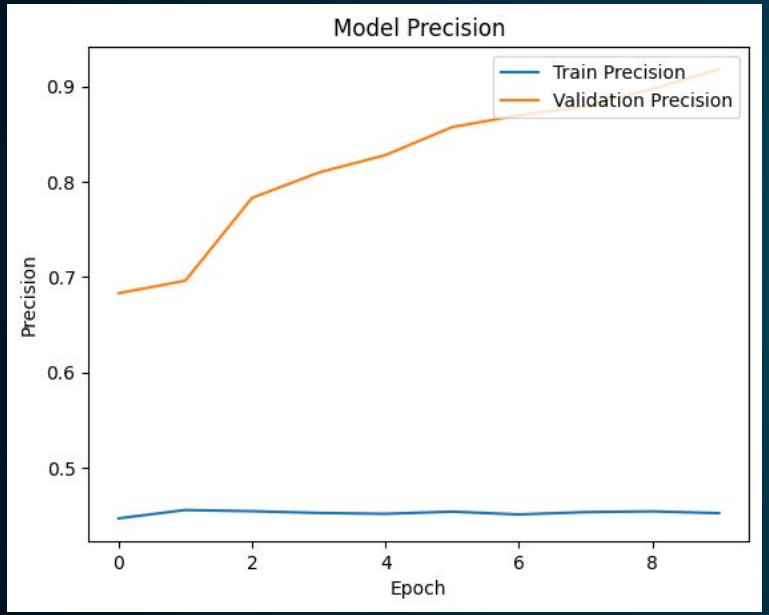
Non-trainable params: 896 (3.50 KB)

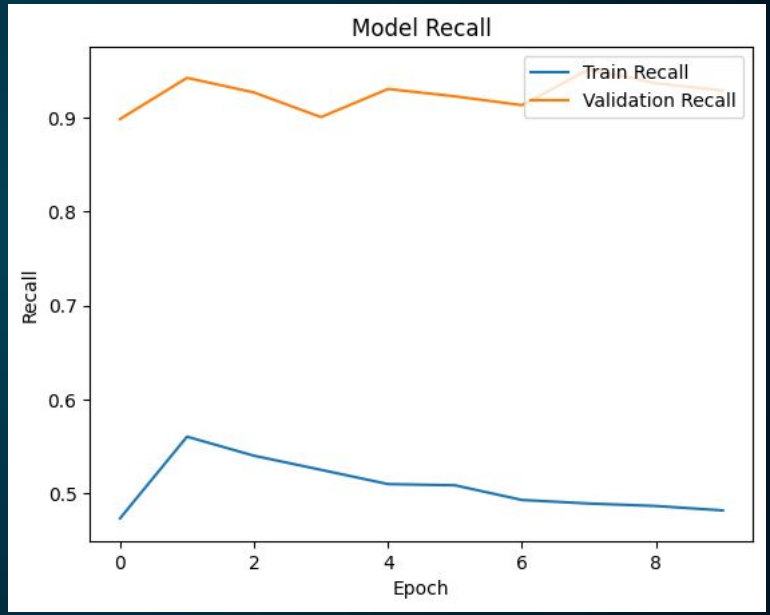
TWO PATHS CNN WITH ATTENTION GATES CNTD.



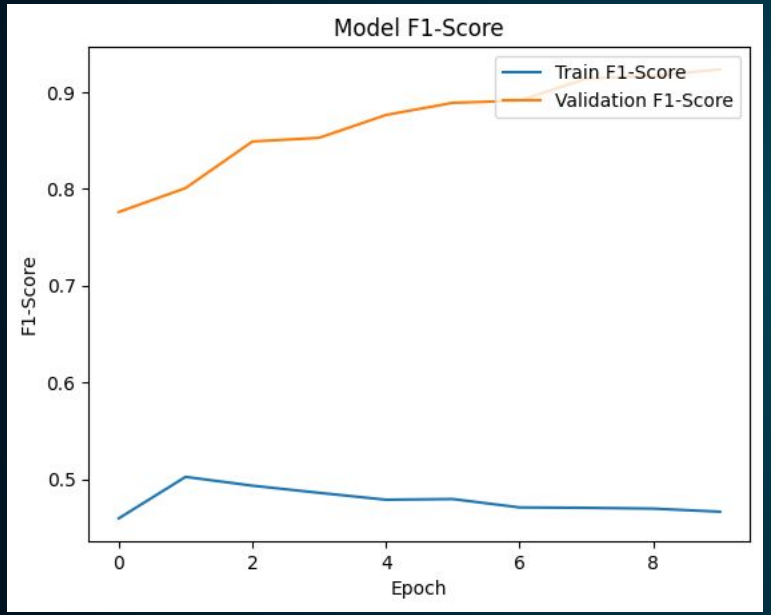


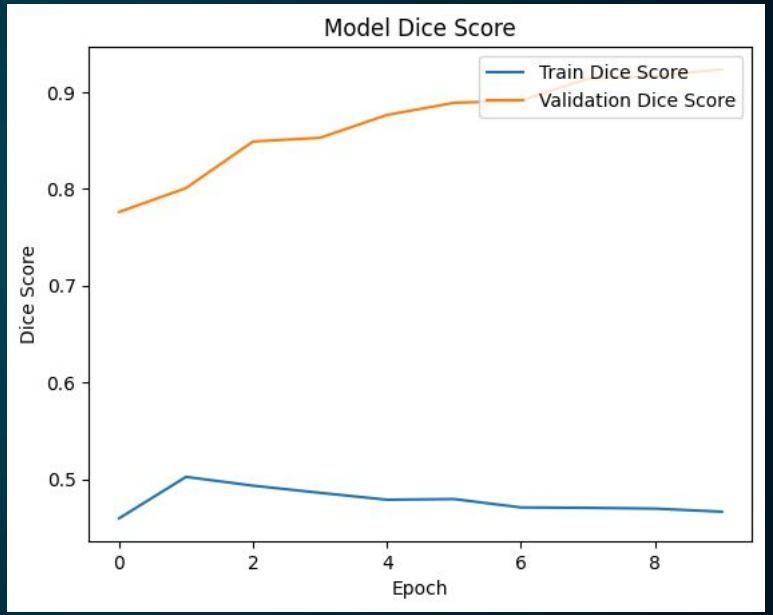
TWO PATHS CNN WITH ATTENTION GATES CNTD.





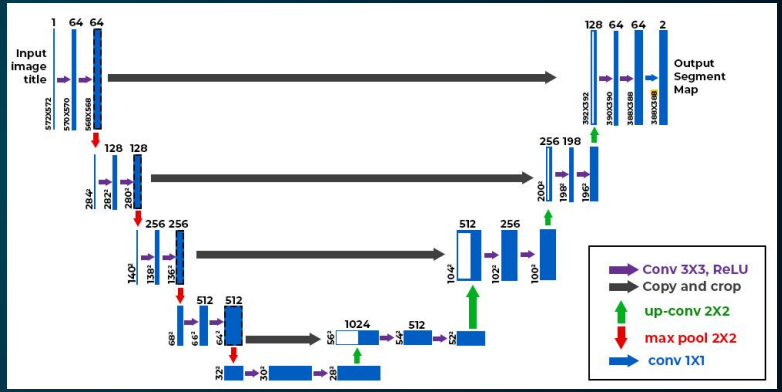
TWO PATHS CNN WITH ATTENTION GATES CNTD.





U-NET

- CNN specifically designed for biomedical image segmentation
- Effective for segmenting regions of interest that are small and have complex shapes
- Introduced in 2015 by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in their paper
- U-NET was based on pixel segmentation but was adjusted to depend on image segmentation instead



U-NET CNTD.

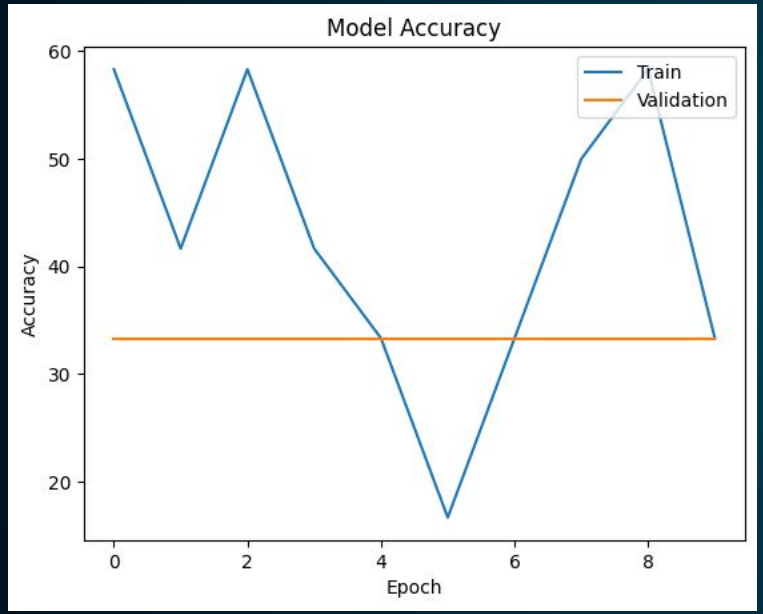
Layer (type)	Output Shape	Param #	Connected to
input_layer_8 (InputLayer)	(None, 224, 224, 4)	0	-
conv2d_96 (Conv2D)	(None, 224, 224, 32)	1,184	input_layer_8[0]…
conv2d_97 (Conv2D)	(None, 224, 224, 32)	9,248	conv2d_96[0][0]
max_pooling2d_14 (MaxPooling2D)	(None, 112, 112, 32)	0	conv2d_97[0][0]
conv2d_98 (Conv2D)	(None, 112, 112, 64)	18,496	max_pooling2d_14…
conv2d_99 (Conv2D)	(None, 112, 112, 64)	36,928	conv2d_98[0][0]
max_pooling2d_15 (MaxPooling2D)	(None, 56, 56, 64)	0	conv2d_99[0][0]
conv2d_100 (Conv2D)	(None, 56, 56, 128)	73,856	max_pooling2d_15…
conv2d_101 (Conv2D)	(None, 56, 56, 128)	147,584	conv2d_100[0][0]
up_sampling2d_10 (UpSampling2D)	(None, 112, 112, 128)	0	conv2d_101[0][0]
concatenate_13 (Concatenate)	(None, 112, 112, 192)	0	up_sampling2d_10… conv2d_99[0][0]
conv2d_102 (Conv2D)	(None, 112, 112, 64)	110,656	concatenate_13[0…
conv2d_103 (Conv2D)	(None, 112, 112, 64)	36,928	conv2d_102[0][0]
up_sampling2d_11 (UpSampling2D)	(None, 224, 224, 64)	0	conv2d_103[0][0]
concatenate_14 (Concatenate)	(None, 224, 224, 96)	0	up_sampling2d_11… conv2d_97[0][0]
conv2d_104 (Conv2D)	(None, 224, 224, 32)	27,680	concatenate_14[0…
conv2d_105 (Conv2D)	(None, 224, 224, 32)	9,248	conv2d_104[0][0]
conv2d_106 (Conv2D)	(None, 224, 224, 1)	33	conv2d_105[0][0]

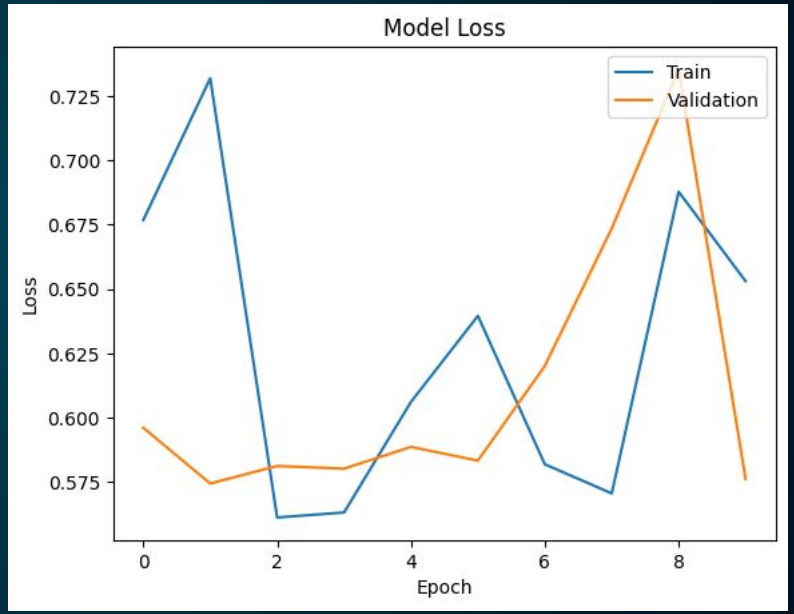
Total params: 471,841 (1.80 MB)

Trainable params: 471,841 (1.80 MB)

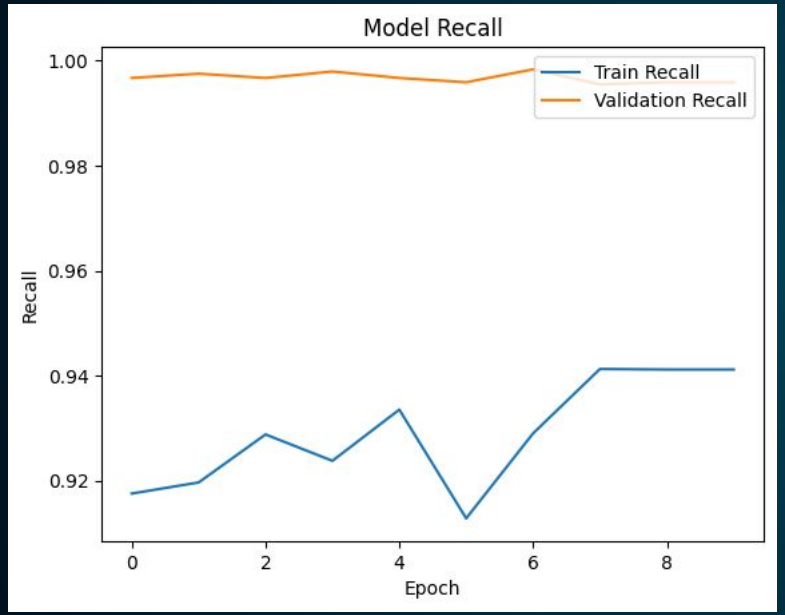
Non-trainable params: 0 (0.00 B)

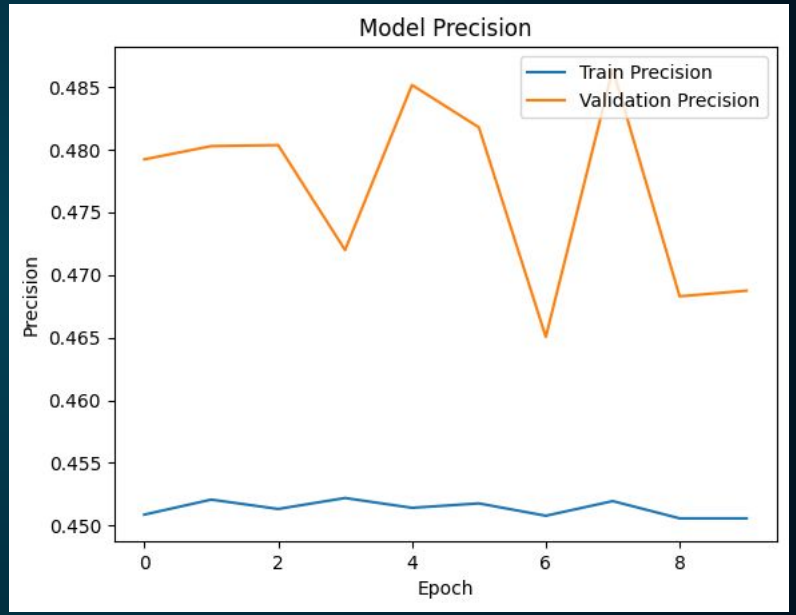
U-NET CNTD.


Model Accuracy
Train
Validation
Accuracy
Epoch
60
50
40
30
20
0
2
4
6
8

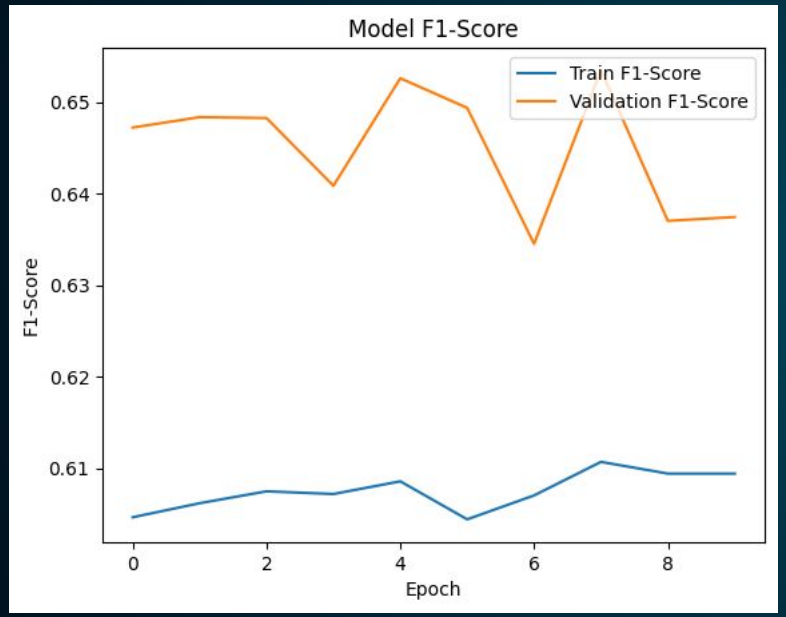

Model Loss
Train
Validation
Loss
Epoch
0.725
0.700
0.675
0.650
0.625
0.600
0.575
0
2
4
6
8

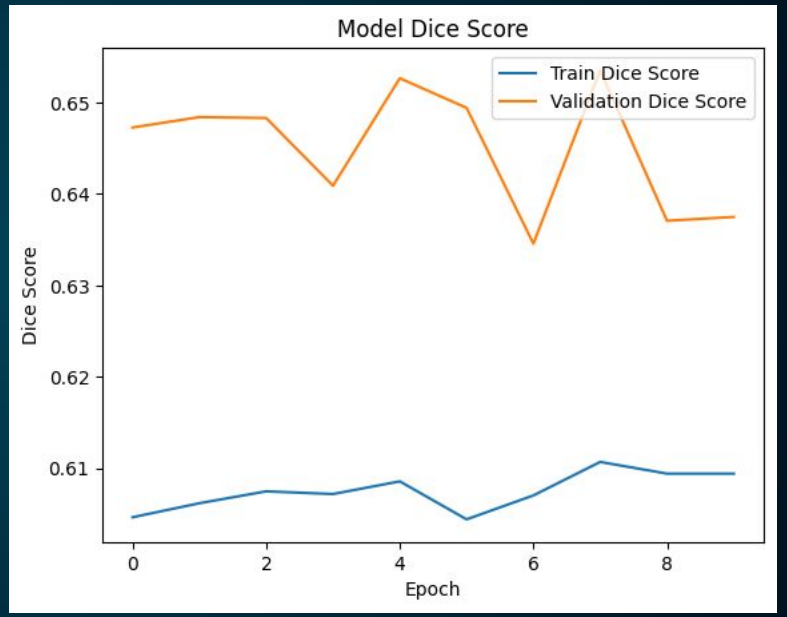
U-NET CNTD.


Model Recall
Train Recall
Validation Recall
Recall
Epoch
1.00
0.98
0.96
0.94
0.92
0
2
4
6
8


Model Precision
Train Precision
Validation Precision
Precision
Epoch
0.485
0.480
0.475
0.470
0.465
0.460
0.455
0.450
0
2
4
6
8

U-NET CNTD.


Model F1-Score
Train F1-Score
Validation F1-Score
F1-Score
0.65
0.64
0.63
0.62
0.61
0
2
4
6
8
Epoch


Model Dice Score
Train Dice Score
Validation Dice Score
Dice Score
0.65
0.64
0.63
0.62
0.61
0
2
4
6
8
Epoch

ENSEMBLE OF TWOPATHS CNN & U-NET

1. Capture the advantages of both
   a. Capture multi-scale & hierarchical features → TwoPaths CNN
   b. Focus on local & global details → TwoPaths CNN
   c. Preserve spatial information → U-Net
   d. Essential for pixel-level tasks → U-Net

2. Better generalization and less overfitting
   a. Each model has different inductive biases
   b. Ensemble may learn different aspects of the data

3. Ensemble done by averaging the predictions by stacking through AdaBoost

ENSEMBLE OF TWOPATHS CNN & U-NET CNTD.

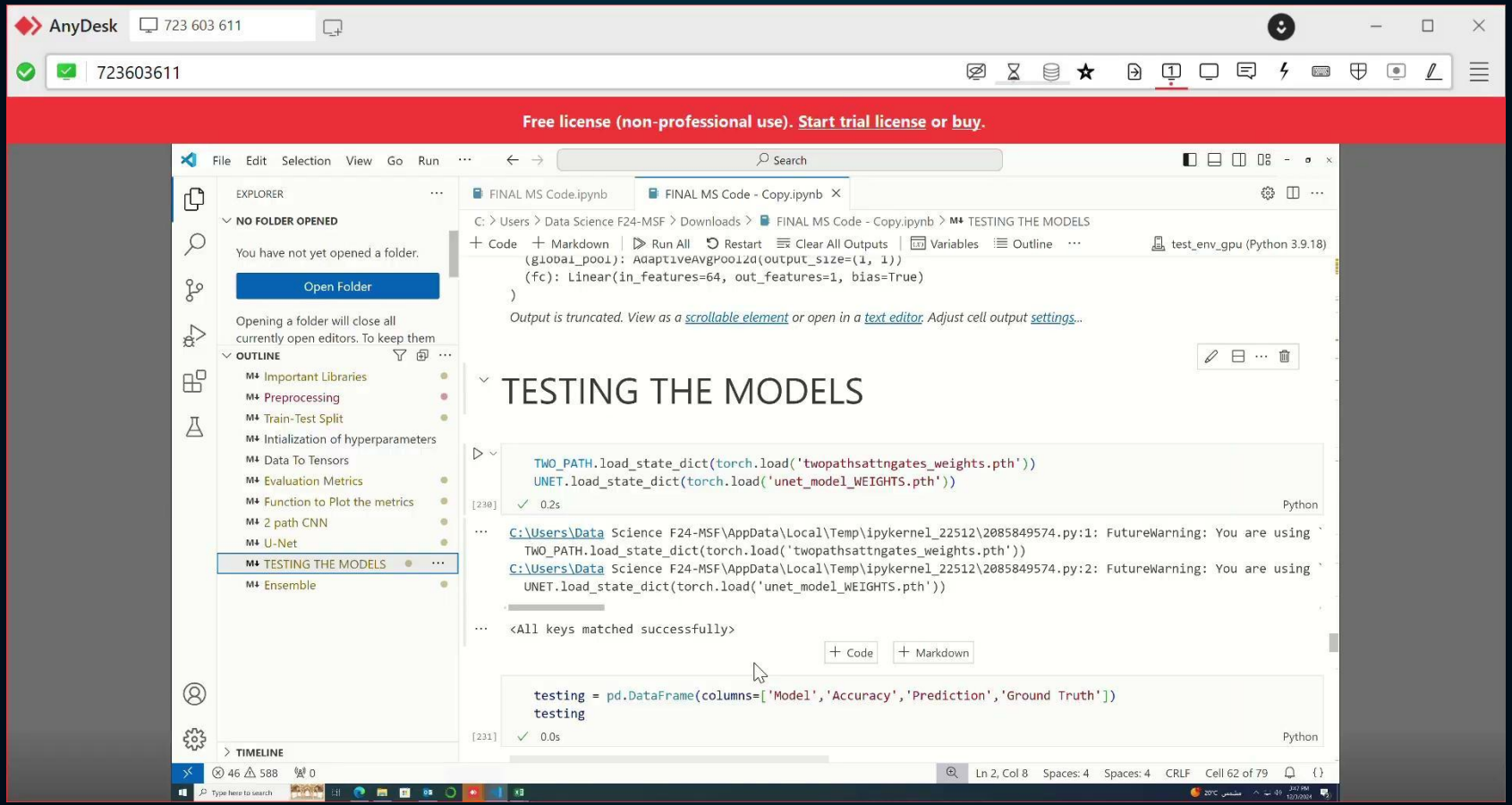
We trained each model separately and then combined their predictions to get better results.
We averaged their outputs to get a better result for the ground truth.
Example:

```
Accuracy is 77.72%.
Majority vote for the image is 1.
Therefore, the MRI scan CONTAINS a brain tumor.
```

```
Ensemble Accuracy with AdaBoost is 74.15%
Precision: 0.80
Recall: 0.72
Dice Score (F1): 0.76
```

03

LIVE DEMO


AnyDesk
723 603 611
723603611
Free license (non-professional use). Start trial license or buy.
File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them
OUTLINE
Important Libraries
Preprocessing
Train-Test Split
Intialization of hyperparameters
Data To Tensors
Evaluation Metrics
Function to Plot the metrics
2 path CNN
U-Net
TESTING THE MODELS
Ensemble
TIMELINE
FINAL MS Code.ipynb
FINAL MS Code - Copy.ipynb
C: > Users > Data Science F24-MSF > Downloads > FINAL MS Code - Copy.ipynb > TESTING THE MODELS
Code Markdown Run All Restart Clear All Outputs Variables Outline
test_env_gpu (Python 3.9.18)
(global_pool): AdaptiveAvgPool2d(output_size=(1, 1))
(fc): Linear(in_features=64, out_features=1, bias=True)
)
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
TESTING THE MODELS
TWO_PATH.load_state_dict(torch.load('twopathsattngates_weights.pth'))
UNET.load_state_dict(torch.load('unet_model_WEIGHTS.pth'))
[230] 0.2s Python
C:\Users\Data Science F24-MSF\AppData\Local\Temp\ipykernel_22512\2085849574.py:1: FutureWarning: You are using `
TWO_PATH.load_state_dict(torch.load('twopathsattngates_weights.pth'))
C:\Users\Data Science F24-MSF\AppData\Local\Temp\ipykernel_22512\2085849574.py:2: FutureWarning: You are using `
UNET.load_state_dict(torch.load('unet_model_WEIGHTS.pth'))
<All keys matched successfully>
Code Markdown
testing = pd.DataFrame(columns=['Model','Accuracy','Prediction','Ground Truth'])
testing
[231] 0.0s Python
46 588 0
Ln 2, Col 8 Spaces: 4 Spaces: 4 CRLF Cell 62 of 79

04

CONCLUSION

CONCLUSION

SKULL STRIPPING

1. Effective preprocessing
2. Identify local changes in intensity of image

PREPROCESSING

Selecting a subset of the data due to computational and RAM issues

MINI-BATCH GD

Enhances model performance over several epochs

TWO PATHS CNN

Baseline model has average performance

CONCLUSION CNTD.

ATTENTION GATES

1. Enhanced the model's performance drastically
2. Focus on the details in the images

U-NET

Baseline model has average performance but less than TwoPaths with attention gates

ENSEMBLE

Combined benefits of both models

ADABOOST

Used for better performance and precision

FUTURE WORK

01

TRYING DIFFERENT MODELS

02

ENHANCE TWOPATHS CNN IN OTHER METHODS

03

TRYING ENSEMBLES WITH DIFFERENT MODELS

TIMELINE AND PROGRESS

TASK	PROGRESS
IMPLEMENT ORIGINAL MODEL	✓
SKULL-STRIPPING	✓
MINI BATCH	✓

TASK	PROGRESS
UNET MODEL	✓
ATTENTION GATES	✓
ENSEMBLE	✓
ADABOOST	✓

TEAM CONTRIBUTION

Farida:

- TwoPaths CNN Baseline Initial Model
- TwoPaths CNN with Attention Gates

Sama:

- TwoPaths CNN Baseline Model Adjustment
- Mini-Batch
- U-Net Model
- Skull Stripping

Mona:

- Ensemble
- Adaboost

THANK YOU!